Extracting Semantic Relatedness from Navigation in a Social Tagging System

Thomas Niebler, Martin Becker, Daniel Zoller, Stephan Doerfel, and Andreas Hotho

No Institute Given

Abstract Semantic relatedness of words has been extracted from a large variety of sources, e.g., tagging data and human navigational paths (e.g., on Wikipedia). While there is strong evidence that navigation in social tagging system is influenced by a semantic component, no attempts have been made yet to extract this component. In this work, we investigate human navigational paths on BibSonomy, a social tagging system, and try to extract information about semantic relatedness. We propose a new method to extract this information and evaluate it against standard datasets as well as a newly constructed dataset more fitting to the BibSonomy vocabulary.

1 Introduction

Semantic relatedness between words or more generally concepts has been extracted from a variety of sources, such as tagging data [3, 10], Wikipedia articles [7, 15] and human navigational paths [**West2009**, 11, 13]. Especially social tagging systems have been in the focus of research, since tagging data provide a valuable internal structure, called a folksonomy [9], which allows for easy extraction of semantic relatedness information between tags [3, 8]. The uses of semantic relatedness measures can be seen in word sense disambiguation algorithms [**niebler2013ecir**], automatic ontology construction[1] and tag recommendation[**bogers2009recommender**, **jaeschke2008tag**]. There has also been a great interest in the pragmatics of user behaviour in social tagging systems, but it only aimed at understanding the reasons behind why a user is assigning tags, not so much on how she navigates the system[**koerner2010thinking**, **niebler2013ecir**].Recently, it could be shown that navigation in social tagging systems is strongly influenced by a semantic component [12]. Furthermore, several methods to extract semantic relatedness from navigation in information networks have been proposed [4, 11, 13, 16]. However, there has been no work up to this point which focused on extracting semantic relatedness from navigation in social tagging systems.

tni

Dieser Satz ist etwas fehl am Platz

In this work, we investigate human navigational paths on the social tagging system BibSonomy and propose a new method to exploit these navigational paths to extract semantic relatedness. We explore our method with many experiments and compare it to several baselines.

The paper is structured as follows: In Section 2, we provide an overview of the experiment and evaluation datasets. Section 3 gives some definitions and describes the process of extracting semantic similarity from navigational paths on Wikipedia. We describe the project setup and prefiltering steps in Section 4. The method presented in Section 3 is applied and extended in Section 5, where we conduct several experiments on the request data from BibSonomy. Finally, Section 6 describes some proposals for future work.

2 Datasets

This section gives an overview over all datasets used in this study. Most datasets are taken from the social tagging system BibSonomy [2] and restricted to include all data from the beginning of the system in 2006 until 01-01-2012. On this day, the login mechanism has been changed and many changes have been incorporated in the page infrastructure. Other datasets are used for comparisons and evaluation.

2.1 BibSonomy user dataset

The user dataset contains a list of all users which registered an account in BibSonomy until 01-01-2012. The list contains the username, the registration date, a flag if there still is a decision to be made on the spammer status of the user and the spammer flag itself. Naturally, we cannot say if a user is a spammer or not if her status is still to be decided. We filtered the list and retained all users where the user decidedly is a nonspammer. See Table 1 for details.

Table 1: All users and the corresponding parts of unclassified and classified users vs. spammers and nonspammers. It doesn't make sense to flag a user as a spammer while her status is still undecided, so we did not differentiate between unflagged users.

	spammer	¬ spammer	Σ
toClassify	249 222		249 222
¬ toClassify	585 018	**17 932**	602 950
			852 172

2.2 BibSonomy folksonomy dataset

The folksonomy dataset contains all tag assignments from all users until 01-01-2012. The users can annotate both bookmarks and publications. The folksonomy dataset in its unfiltered state contains 94 944 813 tag assignments, utilizing 3 504 479 tags which have been assigned to 13 173 227 unique publications and bookmarks by 575 342 users.

2.3 BibSonomy log request dataset

The BibSonomy log files include all HTTP requests (caching is disabled) to the BibSonomy system including common request attributes like IP address, date and referer, as well as a session identifier and a cookie containing the name of the logged-in user. The dataset consists of 160 700 774 entries, which have been generated by 692 007 users on 3 267 393 unique BibSonomy pages. For details see [5].

2.4 WikiGame

The [1] is a competitive navigation game on the articles of Wikipedia. A game instance is given by two randomly drawn Wikipedia pages, the *source* and the *target* pages. Both

[1] `http://www.thewikigame.com`

pages are taken from a complete subgraph of the Wikipedia article network, i.e., each page can be reached from every other page in this subgraph. The most often played game mode aims to reach the target page with the least navigation steps. There are other play modes as well, but we won't focus on them. Users can not use special pages like Search, lists or disambiguation pages. The dataset at hand contains 1 799 015 paths on 360 417 pages from 361 115 games which have been played between Feb, 17th 2009 and Sept, 12th 2011 by 260 095 players. A thorough description of the dataset is given in [13].

2.5 Delicious

Like BibSonomy, Delicious is a social tagging system. Users can share their bookmarks and annotate them with tags. While BibSonomy is research and technically oriented, Delicious is rather focussed on design and computer related topics[17]. The Delicious folksonomy annotates 14 782 752 tags to 118 520 382 bookmarks by 1 951 207 users in 1 026 152 357 tag assignments.

2.6 WS353

WS-353[2] (WordSimilarity-353) [6] consists of 353 pairs of English words and names. Each pair was assigned a relatedness value between 0.0 (no relation) and 10.0 (identical meaning) by 16 raters, denoting the assumed common sense semantic relatedness between two words. Finally, the total rating per pair was calculated as the mean value of each of the 16 users' ratings. This way, WS-353 provides a valuable evaluation base for comparing our concept relatedness scores computed on Wikipedia to an established human generated and validated collection of word pairs.

2.7 Bib100

Bib100 is a dataset specifically to evaluate BibSonomy, because the vocabularies of WS-353 and BibSonomy differ to a great extent and because of this, the evaluation of any data related to BibSonomy against WS-353 is rendered very difficult. The dataset at hand consists of 100 pairs of English words. Like WS-353, each pair was assigned a relatedness value between 0.0 and 10.0, which has been calculated from the ratings of 26 people. Section 5.7 describes the generation process of this dataset.

3 Definitions and methodology

In both [13] and [11], we extracted semantic information from navigational paths on Wikipedia. While in [13], we proposed a method to find cooccurrences between concepts in paths from a navigation game, we refined that method in [11] by expanding it to unconstrained navigation on the public Wikipedia webpage and also introduced an evaluation variation called "binarization" to only take link selection into account instead of link popularity.

In this chapter, we give the definition of a folksonomy and describe a method to extract semantic relatedness from navigational paths on Wikipedia, which we will later on develop to be applied to the BibSonomy folksonomy.

[2] http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html

3.1 Folksonomy

Throughout this paper, we work with folksonomy data. In the following, we will use the definition of a folksonomy from [3].

Definition 1. *A folksonomy is defined as a tuple* $\mathbb{F} := (U, T, R, Y)$*, where* U*,* T *and* R *are finite sets, whose elements are called* users*,* tags *and* resources*, respectively.* Y *is a ternary relation between them, i.e.* $Y \subseteq U \times T \times R$*. A* post *is a triple* (u, T_{ur}, r) *with* $u \in U$*,* $r \in R$ *and a non-empty set* $T_{ur} := \{t \in T | (u, t, r) \in Y\}$*.*

Folksonomies are the data structures underlying *social tagging systems*. In these systems, users collect resources and annotate them with freely chosen keywords, called tags. Examples are BibSonomy[3], for collecting web links and scholarly publications, Delicious[4] (web links), [5] (images), and last.fm[6] (music).

3.2 Extraction of semantic similarity from navigational paths in the WikiGame

In [13], we introduced a method to extract semantic similarity from navigational paths on Wikipedia using a dataset from the .

Given a path $P \in \mathbb{P}$ from the path set $\mathbb{P}$, which is defined as a sequence of pages $(p_1, \ldots, p_n)$, and let $P_{mn} := (p_m, \ldots, p_n) \subset P$ be a subsequence of P, then we create cooccurrence vectors v_i for each page p_i where

$$v_{ij} := \sum_{P \in \mathbb{P}} | \{P_{mn} \subset P | p_i, p_j \in P_{mn}, n - m \leq k\} |$$

for a given window size k. This means, that we create *cooccurrence* or *context vectors* for *webpages*, where the features are the co-visit counts with other webpages in the same paths.

Because every Wikipedia page, which is allowed in the , represents a semantic concept, we can easily substitute the webpages with their corresponding concepts and thus receive term-cooccurrence vectors, which can then be intuitevely and easily compared using the cosine measure, which is the scalar product of two normalized vectors:

$$cossim(i, j) := \frac{\langle v_i, v_j \rangle}{|v_i| \cdot |v_j|} \tag{1}$$

The higher the cosine of two vectors, the more semantically similar are the corresponding concepts, since they share a highly similar context.

3.3 Evaluation of semantic similarity results on a human intuition dataset

We have to evaluate our similarity results from Section 3.2 by comparing them with human intuition of semantic relatedness. For this, we used the WS-353 dataset. This dataset contains 353 pairs of English words and a corresponding similarity score in a

[3] http://www.bibsonomy.org

[4] http://delicious.org

[5] http://www.flickr.com

[6] http://last.fm

range of 0 to 10, where 0 means no similarity and 10 means full similarity. All ratings have been generated by 13-16 people, who were asked to give their idea of relatedness for each of the 353 pairs. The pairs are composed of 437 unique words.

To evaluate our data, we first calculated the overlap of our results with WS-353, i.e. we extracted all pairs where both words are represented by a cooccurrence vector so we can calculate the cosine similarity ((1)). For those common pairs, we then applied the Spearman rank correlation coefficient to the corresponding similarity values. The Spearman rank correlation coefficient ρ for two numerical rankings X and Y is defined as follows:

$$\rho(X,Y) := \frac{Cov(rg_X, rg_Y)}{Var\left[rg_X\right] \cdot Var\left[rg_Y\right]} \tag{2}$$

rg denotes the *rank* of an element in the ranking, i.e. the position of the element in the list, regardless of the actual value by which the list is ordered.

By using the Spearman rank correlation coefficient, we do not have to compare the actual similarity values, but the relation of all pairs to each other. A high absolute correlation value near 1 means almost perfect correlation, i.e. our method yields similar results as humans' intuition would do, whereas a correlation value near 0 means no correlation.

In [13], we left out all pairs for which we calculated similarity values of 0 and excluded them from our evaluation, i.e. marked them as non-matching, because we cannot say if some pairs are dissimilar or have simply not been used in the same context. This way, we received better results than with marking them as found, but purposely excluded bad similarities. We call the exclusion of zero-similarities *optimistic* and the inclusion *pessimistic evaluation*. Throughout this work, we will stay with the pessimistic evaluation.

In [11], we applied the method from Section 3.2 to a dataset of real-life navigation () on Wikipedia and introduced a slight modification to the evaluation process: We replaced the cooccurrences *counts* with *binary flags*, i.e. we do not count how often two pages have been in the same context, but if they cooccurred at all or not. This reduced the impact of huge cooccurrences on the cosine result, while it strengthened that of rare cooccurrences. Overall, this improved our results by quite a margin. Because of this, we always include both the “normal” and binarized evaluation results in this work.

4 Project Setup

As we deal with raw request data, it is necessary to cleanse the data at hand.

4.1 Pre-Filtering of Requests and Tag Assignments

As a first preliminary step, we filtered the request logs and the tag assignment datasets. We did that to establish an easily understandable data base. Because of this, we restricted the data to what a nonspammer would normally see in the system. We excluded spammers, because we want to investigate how normal users would use BibSonomy.

The request logs (in the uncleansed form) contains requests from both unknown and loggedin users, where the latter can again be divided into nonspammers and all the rest. We only retained requests which have been performed by nonspammers before 01-01-2012, have been direct requests (HTTP status code 200) with a requested mimetype from a predefined list (mainly to exclude non-browsing traffic), and have been made inside of BibSonomy (referer is a BibSonomy URL) on retrieval type pages (see Table 2), i.e. pages which return entries from the folksonomy and which actually hold any semantic meaning[7].

As a final step, we extracted the unique requested pages on BibSonomy, which are contained both in the `target` and `referer` fields of a request. We need these unique requested pages for the next step, where we assign a list of tags to them. See Table 3 for detailed numbers after each filtering step. The remaining requests have been generated by 5 756 users. Table 4 shows the page type frequencies of the unique requests.

We also filtered the tag assignments to include only data from nonspammers. All tags not matching the regular expression `^\w+$`, i.e. all non-alphanumeric tags, were removed. Since their tags do not hold any meaning and have been added automatically, we also removed all tag assignments from the bot users `dblp`, `fbw_hannover`, `fbw` and `taggora`. Finally, we counted the tags by occurrence and removed all tag assignments where the tag is not contained in the top 10 000 occurring tags. Table 5 shows a few numbers about the tag assignment dataset in the filtering steps. After filtering, there are 678 542 unique postings left in the tag assignment table, which have been posted 6 028 users.

4.2 Request Tagging

In the request log filtering step, we extracted the unique requested pages. For each of these pages, we want to find the assigned tags, so we have a dictionary which we use in Section 4.3. We also employ the results of this step in the baseline 4.4.3.

We applied simple heuristics on each page type as described in Table 2. Though it is possible to assign tags to most of the requests, there are quite many which cannot be annotated, because of invalid requested users or tags (e.g. `/user/sdhfjkg` and `/user/hotho/frankreich`), wrong hashes (e.g. `/bibtex/nohash`) or non-existent tag assignments, because all tags on a document have been used only once, which makes them disappear after the top10k filtering step in Section 4.1. Table 4 shows how many unique pages there are and how many of them could be assigned with tags.

4.3 Path building

From the remaining requests after the filtering steps, it is now possible to build navigation paths by sorting the requests by time, grouping them by IP and Session ID and finally concatenating them, i.e. connecting two requests if the referer of the latter matches the target of the earlier. This way, we could create 263 373 paths with a mean path length of 2.821.

[7] There are other types of pages like `/settings` or `/homepage`, but we excluded them for now, since there is no intuitive way to describe these pages semantically.

Table 2: The considered page types in the BibSonomy system. All page types are retrieval types, i.e., they return entries in the folksonomy tag assignments.

page type	**description**	**tag assignment strategy**
`/tag/TAG`	contains all posts from all users which have been tagged with `TAG` (e.g. `/tag/web`)	Find all postings which have been tagged with `TAG` and add up all tag assignments.
`/user/USER`	contains all posts from a specified user (e.g. `/user/hotho`)	Aggregate all tags from all bookmarks and publications which this user posted (which might be very many)
`/user/USER/TAG`	contains all posts from a specified user which have been tagged with `TAG` (e.g. `/user/hotho/web`)	This is a combination of the two earlier page type heuristics. From all postings of this user, we only select those which have been assigned the given tag(s). At the end, all tags are again aggregated.
`/bibtex/INTERHASH`	describes a page of a publication not specific to any user	Aggregate all tags from users who posted the publication described by the given INTERHASH
`/bibtex/INTRAHASH`	describes a page of a publication specific to an ommitted user	Aggregate all tags from users who posted the publication described by the given INTRAHASH[8]
`/bibtex/INTERHASH/USER`	describes a page of a publication specific to a user	Aggregate tags only from the given USER who posted the publication described by the given INTERHASH
`/bibtex/INTRAHASH/USER`	describes a page of a publication specific to a user. Links to the same publication as `/bibtex/INTERHASH/USER`	Aggregate tags only from the given USER who posted the publication described by the given INTRAHASH
`/url/INTRAHASH`	describes a page of a bookmark not specific to any user	Aggregate all tags from users who posted the bookmark described by the given INTERHASH

Table 3: Remaining request counts after each filtering step. The fat number denotes the remaining requests after all filtering steps.

Filtered by date	160 700 774
Filtering step	**Remaining requests**
user	4 162 150
status code	3 520 405
mimetype	3 125 976
BibSonomy referer	2 173 522
retrieval pages	**479 471**
Unique visited pages	181 974

Table 4: Page type frequencies of the unique requests. We can see that specific page types like `/user/USER/TAG` and `/bibtex/INTRAHASH/USER` dominate the requests. The third column denotes the number of all request types which could be annotated with at least 1 tag from the top10k tags in the BibSonomy folksonomy.

page type	**req freq**	**¿1 tag**
`/tag/TAG`	12 593	6 691
`/user/USER`	11 814	4 717
`/user/USER/TAG`	70 205	25 093
`/bibtex/INTERHASH`	10 021	8 568
`/bibtex/INTRAHASH`	154	0
`/bibtex/INTERHASH/USER`	684	10
`/bibtex/INTRAHASH/USER`	71 135	45 182
`/url/INTRAHASH`	5 368	3 033
sum	181 974	93 294

Table 5: Remaining tag assignment counts after each filtering step. The fat number denotes the number of remaining assignments after all filtering steps.

Filtered by date	94 944 813
Filtering step	**Remaining tag assignments**
user + bot	2 850 906
regular expression	2 380 242
top10k restriction	**1 993 425**

Figure **??** shows the path length distribution in a scatter plot for all requests and the resulting paths from the minimum transition occurrence experiment in Section 5.6.

4.4 Baseline calculations

To compare the results we received when evaluating the path data, we collected four different baselines. Results for all baselines can be seen in Table 6. All results are evaluated against WS-353 using the Spearman correlation coefficient.

BibSonomy FolkSonomy This baseline is generated by evaluating the semantic properties of the folksonomy described in Section 2.2. We group the tag assignments Y by user and document. For each tag t_i, we now calculate the cooccurrence or context vector v_i as follows:

$$v_{ij} := |\{(u,r) \in U \times R | (u,t_i,r),(u,t_j,r) \in Y\}|$$

Then we calculate the semantic tag similarity between tags t_i and t_j by using the cosine similarity as described in (1).

Delicious FolkSonomy We used the Delicious folksonomy dataset crawled by [17] to serve as a comparison to the BibSonomy folksonomy baseline. We restricted the folksonomy dataset to the top 10 000 tags. After the tag restriction, we end up with 797 796 374 tag assignments from 1 884 280 users on 92 715 855 bookmarks. We calculate tag similarities the same way as in Section 4.4.1.

BibSonomy Requests as Docs This baseline uses the unique, tagged requests from Section 4.2 as documents, which in turn makes it easy to apply the tag similarity calculations described in Section 4.4.1. This baseline serves as a pointer to the intuition that users select interesting resources.

WikiGame Abstracts as Tag Representations We used the abstracts from the visited pages in the dataset as multi-word representations for the corresponding pages, to compare with the assigned tags to a BibSonomy page as a document. For this, we crawled all abstracts, removed stopwords, filtered nouns by using WordNet, limited all abstracts to the top10k words and calculated cosine similarities based on cooccurrence and tf-idf values for all words.

Table 6: Baseline results. Results are given for Spearman's ρ, binary ρ and the number of matchable pairs when evaluated against WS-353

dataset	ρ	ρ_{bin}	**pairs**
BibSonomy folksonomy	0.447	0.445	168
Delicious folksonomy	0.453	0.187	194
BibSonomy ReqAsDocs	0.363	0.335	168
abstracts as tags	0.399	0.073	283

5 Experiments

In this section, we describe the performed experiments with their motivations and results.

5.1 Experiment 1: Direct Application of the Base Method

As a first experiment, we applied the method described in 3.2 directly on the paths generated in 4.3. This means, we calculate context vectors for the *pages* we visited and then compare the similarity of the pages using the cosine measure. After this, we interpret the pages as descriptions of a concept, so we can measure actual semantic similarity between these concepts and compare them e.g. to WS-353 as described in Section 3.3.

Since in BibSonomy, pages do not fill the role of a concept description as easily in Wikipedia (where we only need the title of the page, because a page describes only one concept), we imposed that concept describing role on `/tag/TAG` and `/user/USER/TAG` pages, i.e. the concept *physics* will be represented by e.g. the BibSonomy page `/tag/physics`.

There are now two possibilities to map a word onto a `*/TAG` page: Either we take the whole word as the query or only as a part of the query, e.g. *physics* can be mapped to `/tag/physics` and `/user/USER/physics` (where, of course, we have to cycle through all users which have used this tag) or we also allow partial mappings like `/tag/metaphysics`, which in turn might give us incorrect results[9], but also allows for a greater variety of possibly correct mappings.

If we found several page matches for a given word, e.g. `/tag/physics` and `/user/einstein/physics`, and want to know the similarity between *physics* and *science*, we would calculate the similarities between all page matches of *physics* and all page matches of *science* and take the maximum similarity over all those match pairs.

The results for both the more narrow, stricter case (`/tag/xyz` is the only tag page match for word *xyz*) as well as the broader, more lenient case (`/tag/vwxyz` also matches *xyz*) can be seen in Figure 1.

Table 7: Results for direct application of the base method with window size 4, where pages are characterized by their URLs. Since many of the concept vectors are extremely sparse, there is a high chance that their cosine measure is 0. We calculated both the optimistic and the pessimistic variants, as described in Section 3.3.

approach	ρ	ρ_{bin}	**pairs**
strict	0.134	0.139	156 (24 non-zero)
lenient	0.187	0.182	190 (42 non-zero)
strict optimistic	-0.249	0.220	25
lenient optimistic	0.092	0.023	42

Because both cases do not yield meaningful results, we investigated the reasons for this. Since we compare only `*/TAG` pages, where a specific tag was requested, and many of the WS-353 words are generally not used often in the audience of BibSonomy, these

[9] such as e.g. `/tag/Samstag` when searching for *tag*

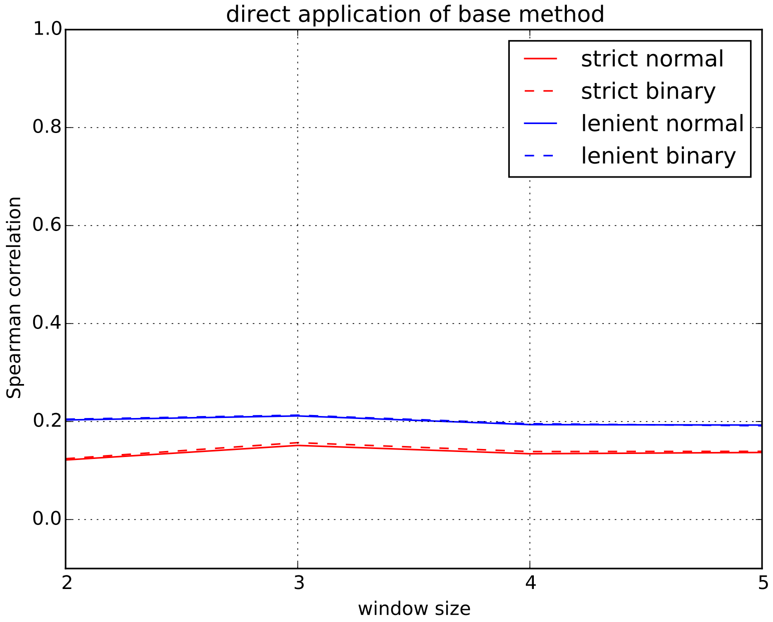


Figure 1: Results for direct application of the base method, where pages are characterized by their URLs. The strict approach only allows direct tag matching (`/tag/physics` for *physics*), while the lenient approach also allows broader matching (`/tag/astrophysics` for *physics*).

pages occur very rarely. The corresponding cooccurrence vectors thus were mostly very sparse, so for most of the WS-353 pairs, we would end up with cosine similarities of 0 between two `*/TAG` vectors and only a few with similarity values ¿ 0. Table 7 shows the results for both the standard pessimistic evaluation (include all similarities in the evaluation) as well as for the optimistic evaluation (exclude pairs with similarity of 0), which have been already described in Section 3.3.

5.2 Experiment 2: Adaptation of WikiGame Method

Because not every visited page was a `*/TAG` page, which can be characterized by the requested tag, we had to somehow characterize other pages like `/user/USER`. We chose to use the tag cloud for each page (see Section 4.2), because all retrieval pages describe either a user, a tag or a document, which in turn are always describable by tags.

In this experiment, we chose the most frequent tag (or randomly one of the most frequent tags, if there was a tie), so we had a concept assigned to each page. This tag served as an equivalent to the title, which is the describing concept of a page in the dataset. We varied the window size k from 2 to 5. Table 8 shows the results of this method.

Table 8: Results for experiment 2, as described in Section 5.2. We chose the most frequent tag for each page as a representative concept. k denotes the window size used in the WikiGame method. ρ and ρ_{bin} are the resulting Spearman correlations for normal and binarized variants.

k	ρ	ρ_{bin}	pairs
2	0.180	0.203	127 (46 non-zero)
3	0.147	0.188	127 (49 non-zero)
4	0.118	0.145	127 (51 non-zero)
5	0.104	0.134	127 (52 non-zero)

5.3 Experiment 3: Tagsets

We extended the method from Section 5.2 to vary the size of the tagsets. Following this, we also modified the way that cooccurrences are calculated.

Tagset size variation Before, we only considered the most frequent tag as a description for a webpage in BibSonomy. In this experiment, we began to vary the size of the assigned tagset on a webpage, so we would have a more verbose description of the page's content. We chose $W := \{1, 10, 20, 50\}$ as possible maximum tagset sizes. For each $w \in W$, we chose the most frequent tags as representations. If there were less than w tags assigned to a webpage, we used all tags.

For this experiment, we had to extend the method to construct cooccurrence vectors. Given two pages p_i and p_j with tagsets $T_i = \{t_{i1}, \dots, t_{in}\}$ and $T_j = \{t_{j1}, \dots, t_{jm}\}$ respectively, we count all elements of the *symmetric cartesian product*

$$T_i \times_s T_j := T_i \times T_j \cup T_j \times T_i \setminus \{(t,t) | t \in T_i \cup T_j\} \tag{3}$$

as cooccurrences. This adaptation, which we call multi-tag cooccurence, is actually a generalization of the base method explained in Section 3.2, because if we set $n = m = 1$, we end up with the original method. An illustration is given in Figure 2. Figure **??** shows the results for the varying tagset sizes with window size k varied between 2 and 5.

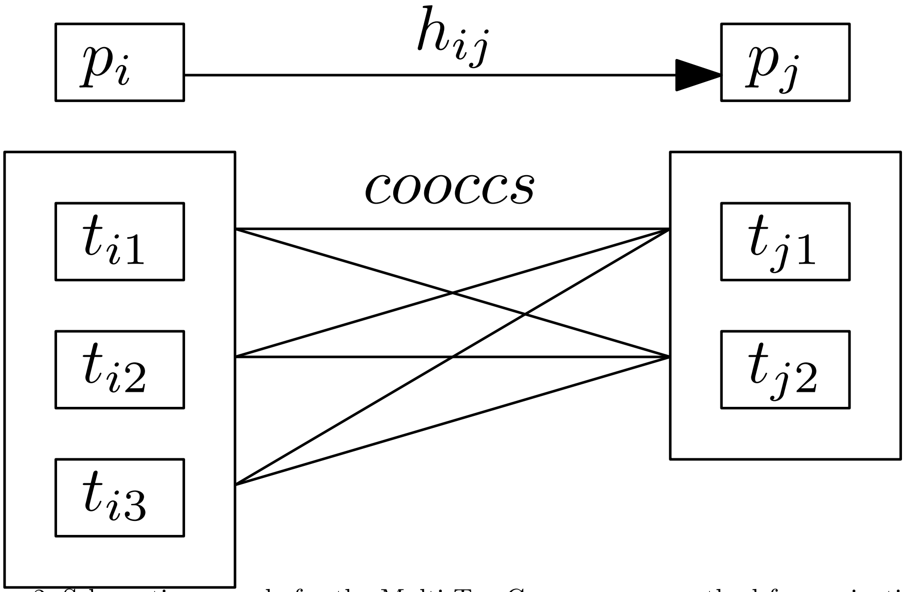
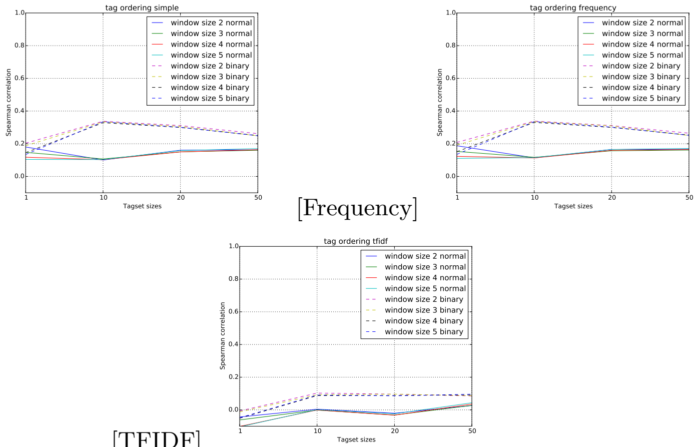


Figure 2: Schematic example for the Multi-Tag-Cooccurrence method for navigational paths. h_{ij} denotes a hypothesis-defined weight for the transition between pages p_i and p_j and is explained more detailed in Section 5.5.

Table 9: The resulting cooccurrence matrix when applying Multi-Tag-Occurrence to the schematic example in Figure 2

	t_{i1}	t_{i2}	t_{i3}	t_{j1}	t_{j2}
t_{i1}	0	0	0	h_{ij}	h_{ij}
t_{i2}	0	0	0	h_{ij}	h_{ij}
t_{i3}	0	0	0	h_{ij}	h_{ij}
t_{j1}	h_{ij}	h_{ij}	h_{ij}	0	0
t_{j2}	h_{ij}	h_{ij}	h_{ij}	0	0

Cooccurrence calculation modifications Until this point, we used a simple counting approach to context vector construction (see Section 3.2). This way, we made no difference if we combined two popular tags or two rarely used tags in the symmetric cartesian product. As can be seen in Table 9, every cooccurrence has been assigned the same weight. But intuitively, often used tags on a page should receive a higher weight than rarely used ones to underline their importance. Because of this, we did not just *count* a cooccurrence, instead we multiplied both occurrences. Since the TF-IDF measure is a widespread alternative to the first order cooccurrence and also assigns a weight to the use of tags, we also calculated the TF-IDF values for all tags. The results can be seen in Figure **??**.



[Simple] [Frequency]

[TFIDF]

Figure 3: Results for the tagset experiment described in Section 5.3. Each picture shows the resulting graphs for varying tagset sizes for the corresponding cooccurrence calculation method, when evaluating the BibSonomy paths on WS-353.

5.4 Experiment 4: Transitions without /user/USER pages

Because a `/user/USER` page is annotated by a big set of tags (see Table 2) which are usually spread across several different topics, we tried to exclude those pages to see if the `/user/USER` pages rather introduce a lot of noise than they are of use. We did so by simply removing all `/user/USER` pages from the rendered paths, e.g.

`/tag/web` → `/user/hotho` → `/user/hotho/web`

became

`/tag/web` → `/user/hotho/web`.

After removing all `/user/USER` requests, there are 245 594 paths left with a mean path length of 1.747. We applied our method to these paths with the simple cooccurrence counting method, the tagsize varied between $\{1, 10, 20, 50\}$ and the window size k varied between 2 and 5. The results can be seen in Figure 4.

In [3], the authors compared results for calculating semantic similarity based on resource, user and tag level. Since the results for path semantics improved a bit when excluding `/user/USER` pages, we could theorize that the same effect of too many aggregated topics on a user personomy[10] shows an effect here.

[10] All posts from a specific user

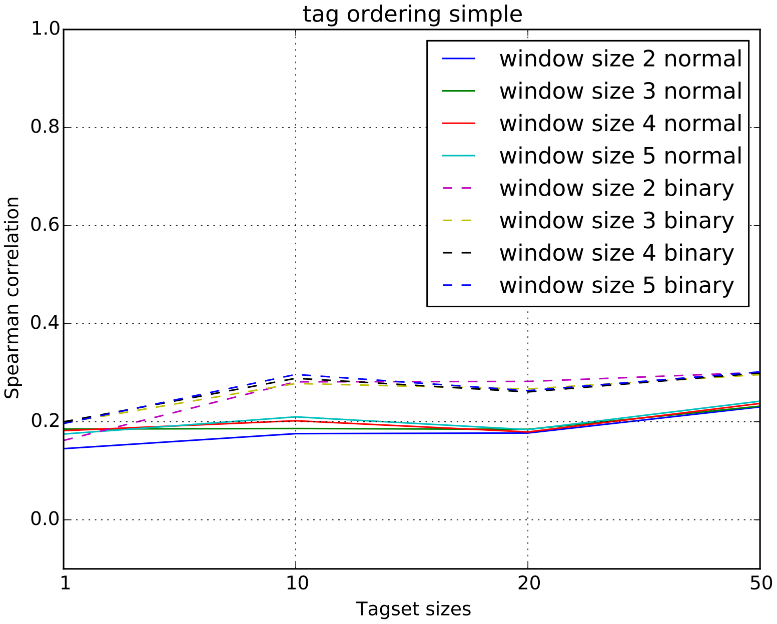


Figure 4: Semantic relatedness results from paths when excluding all `/user/USER` pages.

5.5 Experiment 5: Navigational hypothesis induction into semantic evaluation

Navigational hypotheses about user behaviour are studied in [14] and [**beckerundso**].

In this experiment, we wanted to incorporate navigation hypotheses into the process of semantic relatedness extraction from paths. Hypotheses are represented by transitions matrices, where a matrix component denotes the weight that the corresponding transition is given. This way, it is possible to finely grained encourage navigation between pages or punish it.

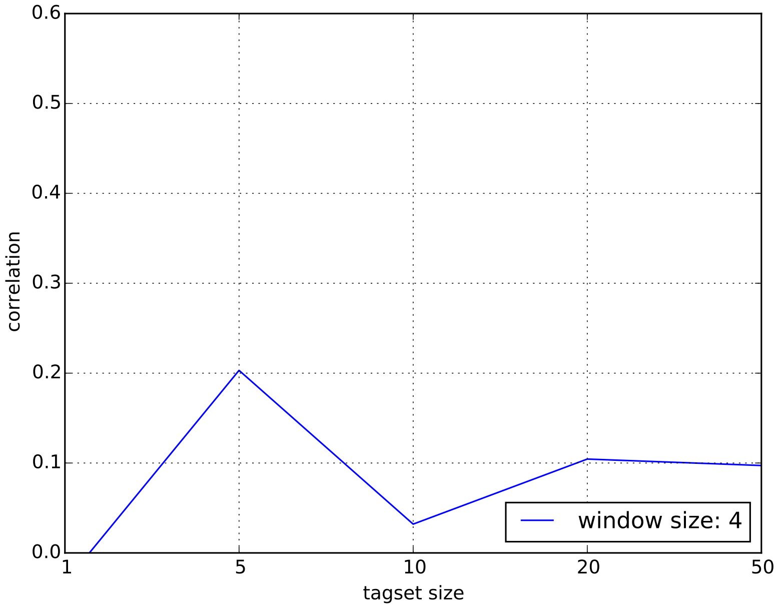
In our case, the transition weight h_{ij} between pages p_i and p_j can easily be included in the symmetric cartesian product calculation process, as Figure 2 and Table 9 show.

Figure 5 shows the results for the "own" hypothesis, where navigation to a navigating user's own pages is encouraged, while navigation to foreign pages is punished, and the "uniform" hypothesis, where each page transition is weighted the same. Window size has been fixed at 4, tagset sizes vary between $\{1, 10, 20, 50\}$.

5.6 Experiment 6: Core Transitions

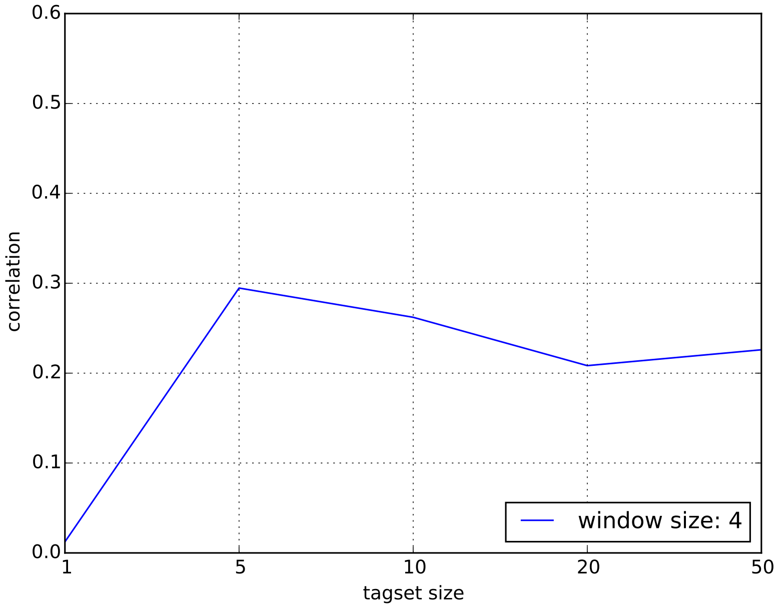
As the semantics extraction method applied both on and yielded very good results, as opposed to the dataset (see [11]), we compared these three datasets with respect to size vs usage ratio. Table 10 shows the sizes, usage counts and the size/usage ratio for each dataset, together with the best achievable result of the application of the binarization method proposed in [11] when evaluated on WS-353.

As we can see, the dataset performs very bad compared to both other datasets. When comparing the features of all three datasets, we can see very quickly that both and have a very low size/usage ratio, while yields a very high ratio. Moreover, consists of transitions which have been observed at least 10 times. The observation gives rise to [normal



evaluation]

[normal



evaluation]

Figure 5: “Own” Hypothesis results, compared with the “uniform” hypothesis. (Old results)

Table 10: Comparison of the Wikipedia navigation datasets used in [11]. The size denotes the number of unique requests, usage describes the total number of requests made in the dataset.

size	usage	size/usage ratio	$\rho_{optimal}$ (pairs)
2.3M	62.5M	0.037	0.728 (236)
14.4M	1 090.2M	0.013	0.709 (288)
2.8M	4.0M	0.7	0.458 (120)

the idea that a lower size/usage ratio might also yield more meaningful results (though we didn't test this on the Wikipedia datasets).

Considering this, we limited the BibSonomy request dataset and removed all transitions which occurred less than a predefined threshold to achieve a similar size/usage ratio or at least a similar effect that fits the idea that low size/usage ratios yield better results. The different dataset sizes can be seen in Table 11. We combined this with the tagset experiment as described in Section 5.3. The results are given in Figure 7. Some data about the evaluation basis are given in Table 11.

Table 11: Data of the core experiment described in Section 5.6. The minimum transition occurrence count *mcnt* is given with the *size/usage ration (s/u)* as well as the average path length of the resulting paths.

mcnt	size	usage	s/u	paths	avg len
1	181 974	479 471	0.380	263 373	2.821
10	10 594	196 285	0.054	88 675	3.214
15	6 164	166 614	0.037	72 306	3.304
20	4 374	150 600	0.029	63 774	3.361

5.7 Experiment 7: Creation of an own evaluation dataset

Throughout our experiments, we used the WS-353 dataset to evaluate our semantic relatedness results on a dataset of human intuition of similarity, as described in Section 3.3. Since the WS-353 dataset covers rather common topics with its choice of evaluation words, it clearly is no ideal dataset to evaluate BibSonomy, since both vocabularies differ substantially. To support this, we compared the word overlap of both datasets with the top100, top1000 and top10k[11] tags from BibSonomy with all words of WS-353, so we could see if the WS-353 words are used rather frequently or infrequently in BibSonomy. The results are shown in Table 12.

Because of this, we decided to create an own, more suitable human evaluation dataset for BibSonomy in a similar fashion as WS-353 was created. For this, we randomly selected a range of words from the top3k tags in BibSonomy, built 100 word pairs from 122 unique words according to our own perceived similarity ascending from unrelated to strongly related and had them rated by 26 different native English speakers on the MicroWorkers platform, similar to Amazon's Mechanical Turk. We cleaned the results by removing raters who gave random or many obviously wrong ratings. The average

[11] i.e. all

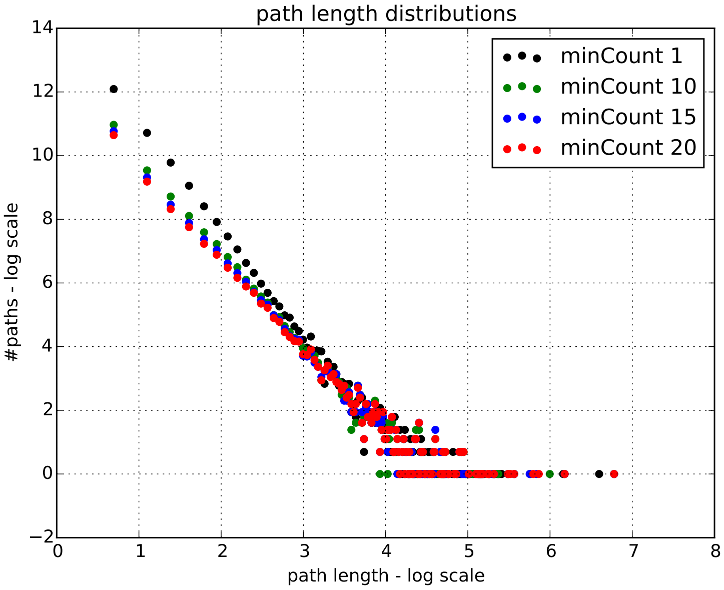


Figure 6: Path length distribution scatter plots for the resulting paths in the minimum transition occurrence experiment. The axes are logarithmically scaled.

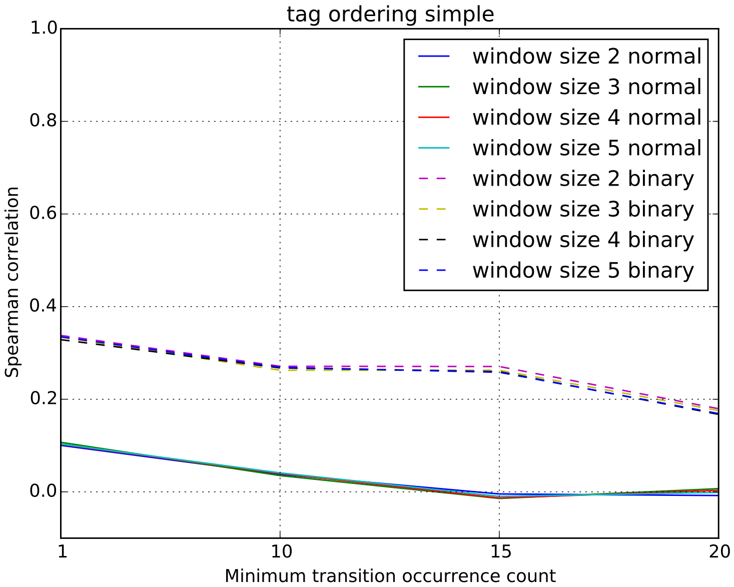


Figure 7: Results for the minimum transitions count experiment. The tagset size was fixed to 10, since that yielded the best results for the simple cooccurrence counting.

interrater agreement is 0.56. Figure 8 shows the mean rating graph with standard deviation graphs around it.

By using this dataset as evaluation basis, our results improved by a medium margin, which shows that, when evaluated on a fitting vocabulary, i) BibSonomy does contain

Table 12: Comparison of different BibSonomy vocabulary subsets with the WS-353 and the Bib100 vocabularies.

BibSonomy subset	**word overlap**	**pair overlap**
WS-353 (353 pairs)		
top 100	18	4
top 1 000	89	36
top 10 000	269	168
Bib100 (100 pairs)		
top 100	14	8
top 1 000	89	70
top 10 000	122	100

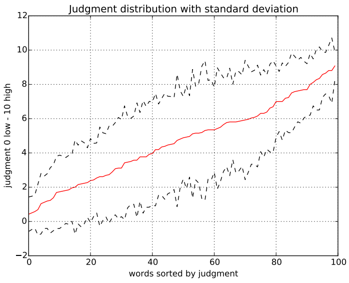


Figure 8: Mean rater scores in Bib100 sorted by ascending score (red graph). The dashed lines denote the standard deviation of the ratings.

semantic information and ii) user navigation on BibSonomy is only partially driven by semantic aspects. The folksonomy baseline results of the evaluation using Bib100 in comparison to WS-353 can be seen in Table 13. Figure **??** shows the evaluation results for experiment 3 in Section 5.3 evaluated against Bib100.

Table 13: Baseline folksonomy scores in comparison when evaluated against WS-353 and Bib100.

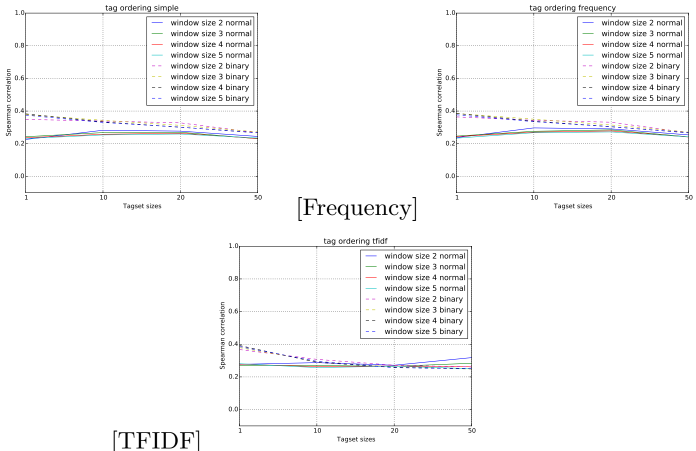
	WS-353	Bib100
BibSonomy folksonomy	0.447	0.600
Delicious folksonomy	0.453	0.640

6 Future work

There is still plenty of work to do. Some work can be done to improve some points in the filtering and execution chain, which is described in Section 6.1. Some points also arise, which haven't been covered in the present work. These are explained in Section 6.2.

6.1 Improvements

In Section 4.3, we could also build navigation trees and extend the semantics extraction method on these trees, which currently only works on paths. This might also add more



[Simple] [Frequency]

[TFIDF]

Figure 9: Results for the tagset experiment described in Section 5.3. Each picture shows the resulting graphs for varying tagset sizes for the corresponding cooccurrence calculation method, when evaluating the BibSonomy paths with Bib100 data.

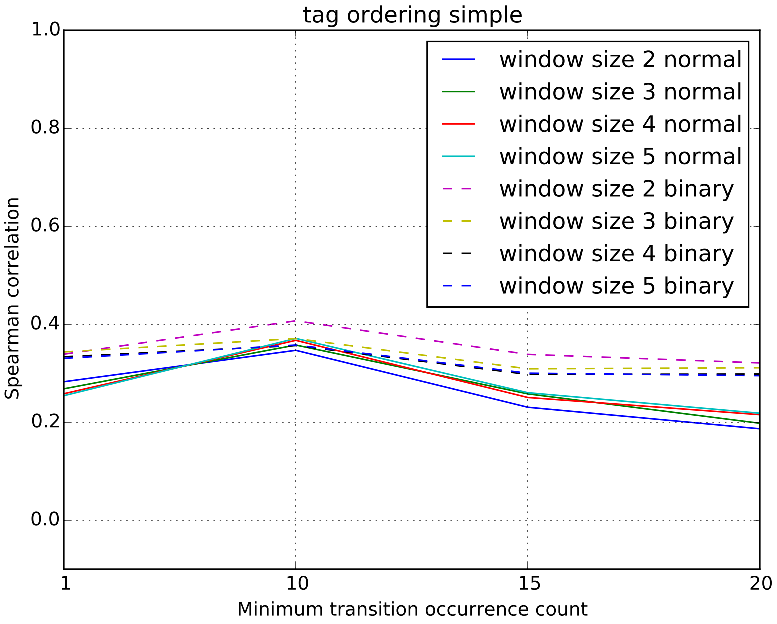


Figure 10: Results for the minimum transitions count experiment, when evaluated against Bib100

context for paths which have possibly not been identified fully because of multitab browsing. It might also be possible to include a time limit on a session, so that we might end up with even more, but shorter paths, which have been created only over time and accidentally started, where another left off.

In Section 4.2, it could help if we only considered parts of the assigned tags, since most times, not all tags are important or even representative for the visited page in its path context. We could just keep the documents instead of a taglist, so we can eliminate uninteresting documents while interpreting a path.

Section 5.3.2 is a great place for variations. There are endless ways to think of a weighing function for the symmetric cartesian product, such as taking the minimum occurrence of both words in a word pair.

6.2 Not covered

Overall, it probably would help, if we had more data and extended the timeframe to include 2012 or even 2013. Also, we could try to identify users with a positive effect on navigational semantics and even construct a navigation behaviour which actually benefits our method.

We could also possibly exploit navigational patterns (i.e. subsets) to extract semantic relatedness information. That might be possible if we utilize neural networks.

bibliography